```
# UCSC Extension
# DBDA.X409.(10) MySQL and Oracle Database for Developers and
Designers
#
# Final Test
# Part 2
# Dynamic SQL
#
# Cheng Fei
# 11/17/2022
#
# Pass in the score table in the studentdb database, extract/compute
5 columns:
# STUDENT_ID| EVENT_ID| MAX_SCORE | MIN_SCORE | AVG_SCORE and
# return them in a new table.

# Change the current database to studentdb.
USE studentdb;

# @_CREATE_PROCEDURE_
# Create a new table based on the score table in the studentdb
database.
# Require the input in_table_name parameter to be "score".
DELIMITER $
DROP PROCEDURE IF EXISTS compute_score_statistics$
DROP TABLE IF EXISTS score_stat$
CREATE PROCEDURE compute_score_statistics(IN in_table_name
VARCHAR(10),

                  # the input table name

                  OUT out_status_msg VARCHAR(10),
                                          # return  “Success”/
"Failed"
                                          OUT out_status_code INT,
                                          # return as “0” as success
and -1 as failed
                                          OUT out_output_table
VARCHAR(10)
                                          # return the tablename
after storing the results into the table
                                          )
BEGIN
        DECLARE v_sql VARCHAR(1000);
    DECLARE l_table_name VARCHAR(10);
        BEGIN
                SET @l_table_name = in_table_name;
        # Execute table creation.
                SET @v_sql := concat('CREATE TABLE score_stat AS ',
                                                        'SELECT
student_id, event_id, ',
                             'MAX(score) max_score, ',
                             'MIN(score) min_score, ',
                             'AVG(score) avg_score ',
```

```
                            'FROM ', @l_table_name, ' ',
                            'GROUP BY student_id, event_id');
                SELECT @v_sql script;
                PREPARE stmt FROM @v_sql;
        EXECUTE stmt;
        # Check whether the table is created successfully.
        SET out_output_table = 'score_stat';
        IF (out_output_table IS NULL) THEN
                        SET out_status_code := -1;
            SET out_status_msg = 'Failed';
        ELSE
                        SET out_status_code := 0;
            SET out_status_msg = 'Success';
        END IF;
    END;
END$
DELIMITER ;

# Commit changes.
COMMIT;
```